

Middlesex University Research Repository

An open access repository of
Middlesex University research

<http://eprints.mdx.ac.uk>

Alechina, Natasha, Logan, Brian, Nguyen, Hoang Nga and Raimondi, Franco ORCID
logoORCID: <https://orcid.org/0000-0002-9508-7713> (2014) Decidable model-checking for a
resource logic with production of resources. ECAI 2014: 21st European Conference on Artificial
Intelligence, 18-22 August 2014, Prague, Czech Republic, including prestigious applications of
artificial intelligence (PAIS-2014) proceedings. In: 21st European Conference on Artificial
Intelligence (ECAI-2014), 18-22 Aug 2014, Prague, Czech Republic. ISBN 9781614994183.
ISSN 0922-6389 [Conference or Workshop Item] (doi:10.3233/978-1-61499-419-0-9)

Published version (with publisher's formatting)

This version is available at: <https://eprints.mdx.ac.uk/15136/>

Copyright:

Middlesex University Research Repository makes the University's research available electronically.

Copyright and moral rights to this work are retained by the author and/or other copyright owners unless otherwise stated. The work is supplied on the understanding that any use for commercial gain is strictly forbidden. A copy may be downloaded for personal, non-commercial, research or study without prior permission and without charge.

Works, including theses and research projects, may not be reproduced in any format or medium, or extensive quotations taken from them, or their content changed in any way, without first obtaining permission in writing from the copyright holder(s). They may not be sold or exploited commercially in any format or medium without the prior written permission of the copyright holder(s).

Full bibliographic details must be given when referring to, or quoting from full items including the author's name, the title of the work, publication details where relevant (place, publisher, date), pagination, and for theses or dissertations the awarding institution, the degree type awarded, and the date of the award.

If you believe that any material held in the repository infringes copyright law, please contact the Repository Team at Middlesex University via the following email address:

eprints@mdx.ac.uk

The item will be removed from the repository while any claim is being investigated.

See also repository copyright: re-use policy: <http://eprints.mdx.ac.uk/policies.html#copy>

Decidable Model-Checking for a Resource Logic with Production of Resources

Natasha Alechina¹ and Brian Logan¹ and Hoang Nga Nguyen¹ and Franco Raimondi²

Abstract. Several logics for expressing coalitional ability under resource bounds have been proposed and studied in the literature. Previous work has shown that if only consumption of resources is considered or the total amount of resources produced or consumed on any path in the system is bounded, then the model-checking problem for several standard logics, such as Resource-Bounded Coalition Logic (RB-CL) and Resource-Bounded Alternating-Time Temporal Logic (RB-ATL) is decidable. However, for coalition logics with unbounded resource production and consumption, only some undecidability results are known. In this paper, we show that the model-checking problem for RB-ATL with unbounded production and consumption of resources is decidable.

1 INTRODUCTION

Alternating Time Temporal Logic (ATL) [2] is widely used in verification of multi-agent systems. ATL can express properties related to coalitional ability, for example one can state that a group of agents A has a strategy (a choice of actions) such that whatever the actions by the agents outside the coalition, any computation of the system generated by the strategy satisfies some temporal property. A number of variations on the semantics of ATL exist: agents may have perfect recall or be memoryless, and they may have full or partial observability. In the case of fully observable models and memoryless agents, the model checking problem for ATL is polynomial in the size of the model and the formula, while it is undecidable for partially observable models where agents have perfect recall [3]. Additionally, even in the simple case of fully observable models and memoryless agents, the complexity increases substantially if the model checking problem takes into account models with *compact* (implicit) representations [3].

In this paper, we consider an extension of perfect recall, fully observable ATL where agents produce and consume resources. The properties we are interested in are related to coalitional ability under resource bounds. Instead of asking whether a group of agents has a strategy to enforce a certain temporal property, we are asking whether the group has a strategy which can be executed under a certain resource bound (e.g., if the agents have at most b_1 units of resource r_1 and b_2 units of resource r_2). Clearly, some actions may no longer be used as part of the strategy if their cost exceeds the bound. There are several ways in which the precise notion of the cost of a strategy can be defined. For example, one can define it as the maximal cost of any path (computation of the system) generated by the

strategy, where the cost of a path is the sum of resources produced and consumed by actions on the path. We have chosen a different definition which says that a strategy has a cost at most b if for every path generated by the strategy, every *prefix* of the path has cost at most b . This means that a strategy cannot, for example, start with executing an action that consumes more than b resources, and then ‘make up’ for this by executing actions that produce enough resources to bring the total cost of the path under b . It is however possible to first produce enough resources, and then execute an action that costs more than b , ensuring the cost of the path is less than b .

There are also many choices for the precise syntax of the logic and the truth definitions of the formulas. For example, in [4] several versions are given, intuitively corresponding to considering resource bounds both on the coalition A and the rest of the agents in the system, considering a fixed resource endowment of A in the initial state which affects their endowment after executing some actions, etc. Our logic is closest (but not identical) to \mathcal{L}_{RAL} with perfect recall, resource-flat, only proponents resource-restricted, and with finitary semantics defined in [4]. Decidability of the model-checking problem for this version of \mathcal{L}_{RAL} was stated as an open problem in [4]. In [6, 7] a different syntax and semantics are considered, involving resource endowment of the whole system when evaluating a statement concerning a group of agents A . As observed in [4], subtle differences in truth conditions for resource logics result in the difference between decidability and undecidability of the model-checking problem. In [4], undecidability for several versions of the logics is proved. The only decidable cases considered in [4] are an extension of Computation Tree Logic (CTL) [5] with resources (essentially one-agent ATL) and the version where on every path only a fixed finite amount of resources can be produced. Similarly, [6] gives a decidable logic PRB-ATL (Priced Resource-Bounded ATL) where the total amount of resources in the system has a fixed bound. The model-checking algorithm for PRB-ATL runs in time polynomial in the size of the model and exponential in the number of resources and the resource bound on the system. In [7] an EXPTIME lower bound in the number resources is shown.

2 SYNTAX AND SEMANTICS OF RB±ATL

The logic RB-ATL was introduced in [1]. Here we generalise the definitions from [1] to allow for production as well as consumption of resources. To avoid confusion with the consumption-only version of the logic from [1], we refer to RB-ATL with production and consumption of resources as RB±ATL.

Let $Agt = \{a_1, \dots, a_n\}$ be a set of n agents and $Res = \{res_1, \dots, res_r\}$ be a set of r resources, Π denote a set of propositions and $B = \mathbb{N}_\infty^r$ denote a set of resource bounds where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$.

¹ School of Computer Science, University of Nottingham, UK email: {nza,bsl,hnn}@cs.nott.ac.uk

² Department of Computer Science, Middlesex University, UK, email: f.raimondi@mdx.ac.uk

Formulas of $\text{RB}\pm\text{ATL}$ are defined by the following syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle\langle A^b \rangle\rangle \bigcirc \varphi \mid \langle\langle A^b \rangle\rangle \Box \varphi \mid \langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$$

where $p \in \Pi$ is a proposition, $A \subseteq \text{Agt}$, and $b \in B$ is a resource bound. Here, $\langle\langle A^b \rangle\rangle \bigcirc \varphi$ means that a coalition A can ensure that the next state satisfies φ under resource bound b . $\langle\langle A^b \rangle\rangle \Box \varphi$ means that A has a strategy to make sure that φ is always true, and the cost of this strategy is at most b . Similarly, $\langle\langle A^b \rangle\rangle \varphi \mathcal{U} \psi$ means that A has a strategy to enforce ψ while maintaining the truth of φ , and the cost of this strategy is at most b .

We extend the definition of concurrent game structure with resource consumption and production.

Definition 1. A resource-bounded concurrent game structure (RB-CGS) is a tuple $M = (\text{Agt}, \text{Res}, S, \Pi, \pi, \text{Act}, d, c, \delta)$ where:

- Agt is a non-empty set of n agents, Res is a non-empty set of r resources and S is a non-empty set of states;
- Π is a finite set of propositional variables and $\pi : \Pi \rightarrow \wp(S)$ is a truth assignment which associates each proposition in Π with a subset of states where it is true;
- Act is a non-empty set of actions which includes *idle*, and $d : S \times \text{Agt} \rightarrow \wp(\text{Act}) \setminus \{\emptyset\}$ is a function which assigns to each $s \in S$ a non-empty set of actions available to each agent $a \in \text{Agt}$. For every $s \in S$ and $a \in \text{Agt}$, *idle* $\in d(s, a)$. We denote joint actions by all agents in Agt available at s by $D(s) = d(s, a_1) \times \dots \times d(s, a_n)$;
- $c : S \times \text{Agt} \times \text{Act} \rightarrow \mathbb{Z}^r$ is a partial function which maps a state s , and agent a and an action $\alpha \in d(s, a)$ to a vector of integers where the integer in position i indicates consumption or production of resource res_i by the action (positive value for consumption and negative value for production). We stipulate that $c(s, a, \text{idle}) = \vec{0}$ for all $s \in S$ and $a \in \text{Agt}$ where $\vec{0} = 0^r$.
- $\delta : (s, \sigma) \mapsto S$ is a function that for every $s \in S$ and joint action $\sigma \in D(s)$ gives the state resulting from executing σ in s .

Given a RB-CGS M , we denote the set of all infinite sequences of states (computations) by S^ω and the set of non-empty finite sequences of states by S^+ . For a computation $\lambda = s_0 s_1 \dots \in S^\omega$, we use the notation $\lambda[i] = s_i$ and $\lambda[i, j] = s_i \dots s_j \forall j \geq i \geq 0$.

Given a RB-CGS M and a state $s \in S$, a joint action by a coalition $A \subseteq \text{Agt}$ is a tuple $\sigma_A = (\sigma_a)_{a \in A}$ such that $\sigma_a \in d(s, a)$. The set of all joint actions for A at state s is denoted by $D_A(s)$. Given a joint action by the grand coalition $\sigma \in D(s)$, σ_A denotes the joint action executed by A : $\sigma_A = (\sigma_a)_{a \in A}$. The set of all possible outcomes of a joint action $\sigma_A \in D_A(s)$ at state s is:

$$\text{out}(s, \sigma_A) = \{s' \in S \mid \exists \sigma' \in D(s) : \sigma_A = \sigma'_A \wedge s' = \delta(s, \sigma')\}$$

The cost of a joint action $\sigma_A \in D_A(s)$ is defined as $\text{cost}(s, \sigma_A) = \sum_{a \in A} c(s, a, \sigma_a)$.

Given a RB-CGS M , a strategy for a coalition $A \subseteq \text{Agt}$ is a mapping $F_A : S^+ \rightarrow \text{Act}$ such that, for every $\lambda s \in S^+$, $F_A(\lambda s) \in D_A(s)$. A computation $\lambda \in S^\omega$ is consistent with a strategy F_A iff, for all $i \geq 0$, $\lambda[i+1] \in \text{out}(\lambda[i], F_A(\lambda[0, i]))$. We denote by $\text{out}(s, F_A)$ the set of all consistent computations λ of F_A that start from s .

In the sequel, we use the usual point-wise notation for vector comparison and addition. In particular, $(b_1, \dots, b_r) \leq (d_1, \dots, d_r)$ iff $b_i \leq d_i \forall i \in \{1, \dots, r\}$, and $(b_1, \dots, b_r) + (d_1, \dots, d_r) = (b_1 + d_1, \dots, b_r + d_r)$.

Given a bound $b \in B$, a computation $\lambda \in \text{out}(s, F_A)$ is b -

consistent with F_A iff, for every $i \geq 0$,

$$\sum_{j=0}^i \text{cost}(\lambda[j], F_A(\lambda[0, j])) \leq b$$

Note that this definition implies that the cost of every prefix of the computation is below b .

The set of all b -consistent computations of F_A starting from state s is denoted by $\text{out}(s, F_A, b)$. F_A is a b -strategy iff $\text{out}(s, F_A) = \text{out}(s, F_A, b)$ for any state s .

Given a RB-CGS M , a state s of M , the truth of a $\text{RB}\pm\text{ATL}$ formula φ with respect to M and s is defined inductively on the structure of φ as follows (the atomic case and the Boolean connectives are defined in the standard way):

- $M, s \models \langle\langle A^b \rangle\rangle \bigcirc \phi$ iff \exists b -strategy F_A such that for all $\lambda \in \text{out}(s, F_A)$: $M, \lambda[1] \models \phi$;
- $M, s \models \langle\langle A^b \rangle\rangle \Box \phi$ iff \exists b -strategy F_A such that for all $\lambda \in \text{out}(s, F_A)$ and $i \geq 0$: $M, \lambda[i] \models \phi$; and
- $M, s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ iff \exists b -strategy F_A such that for all $\lambda \in \text{out}(s, F_A)$, $\exists i \geq 0$: $M, \lambda[i] \models \psi$ and $M, \lambda[j] \models \phi$ for all $j \in \{0, \dots, i-1\}$.

Since the infinite resource bound version of $\text{RB}\pm\text{ATL}$ modalities correspond to the standard ATL modalities, we will write $\langle\langle A^\infty \rangle\rangle \bigcirc \phi$, $\langle\langle A^\infty \rangle\rangle \phi \mathcal{U} \psi$, $\langle\langle A^\infty \rangle\rangle \Box \phi$ as $\langle\langle A \rangle\rangle \bigcirc \phi$, $\langle\langle A \rangle\rangle \phi \mathcal{U} \psi$, $\langle\langle A \rangle\rangle \Box \phi$, respectively. When the context is clear, we will sometimes write $s \models \phi$ instead of $M, s \models \phi$.

Note that although we only consider infinite paths, the condition that the *idle* action of cost $\vec{0}$ is always available makes the model-checking problem easier (we only need to find a strategy with a finite prefix under bound b to satisfy formulas of the form $\langle\langle A^b \rangle\rangle \bigcirc \phi$ and $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$, and then the strategy can make the *idle* choice forever). This makes our logic closer to the finitary semantics in [4].

As an example of the expressivity of the logic, consider the model in Figure 1 with two agents a_1 and a_2 and two resources r_1 and r_2 . Let us assume that $c(s_I, a_1, \alpha) = \langle -2, 1 \rangle$ (action α produces 2 units of r_1 and consumes one unit of r_2), $c(s, a_2, \beta) = \langle 1, -1 \rangle$ and $c(s, a_1, \gamma) = \langle 5, 0 \rangle$. Then agent a_1 on its own has a strategy to enforce a state satisfying p under resource bound of 3 units of r_1 and 1 unit of r_2 ($M, s_I \models \langle\langle a_1 \rangle\rangle^{(3,1)} \top \mathcal{U} p$): a_1 has to select action α in s_I which requires it to consume one unit of r_2 but produces two units of r_1 , and then action γ in s that requires 5 units of r_1 which is now within the resource bound since the previous action has produced 2 units. All outcomes of this strategy lead to s' where p holds. After this, a_1 has to select *idle* forever, which does not require any resources. Any smaller resource bound is not sufficient. However, both agents have a strategy to enforce the same outcome under a smaller resource bound of just one unit of r_2 ($M, s_I \models \langle\langle a_1, a_2 \rangle\rangle^{(0,1)} \top \mathcal{U} p$): agent a_2 needs to select β in s until the agents have gone through the loop between s_I and s four times and accumulated enough of resource r_1 to enable agent a_1 to perform γ in s .

3 MODEL CHECKING $\text{RB}\pm\text{ATL}$

The model-checking problem for $\text{RB}\pm\text{ATL}$ is the question whether for a given RB-CGS structure M , a state s in M and an $\text{RB}\pm\text{ATL}$ formula ϕ , $M, s \models \phi$. In this section we prove the following theorem:

Theorem 1. The model-checking problem for $\text{RB}\pm\text{ATL}$ is decidable.

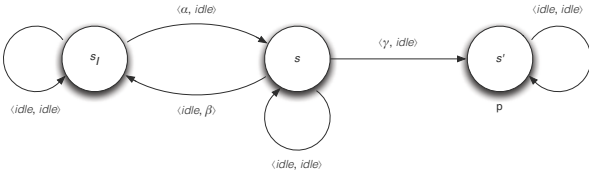


Figure 1. An example with consumption and production of resources.

To prove decidability, we give an algorithm which, given a structure $M = (Agt, Res, S, \Pi, \pi, Act, d, c, \delta)$ and a formula ϕ , returns the set of states $[\phi]_M$ satisfying ϕ : $[\phi]_M = \{s \mid M, s \models \phi\}$ (see Algorithm 1).

Algorithm 1 Labelling ϕ

```

function RB $\pm$ ATL-LABEL( $M, \phi$ )
  for  $\phi' \in Sub(\phi)$  do
    case  $\phi' = p, \neg\psi, \psi_1 \wedge \psi_2,$ 
       $\langle\langle A \rangle\rangle \bigcirc \psi, \langle\langle A \rangle\rangle \psi_1 \mathcal{U} \psi_2, \langle\langle A \rangle\rangle \Box \psi$ 
      standard, see [2]
    case  $\phi' = \langle\langle A^b \rangle\rangle \bigcirc \psi$ 
       $[\phi']_M \leftarrow Pre^b(A, [\psi]_M)$ 
    case  $\phi' = \langle\langle A^b \rangle\rangle \psi_1 \mathcal{U} \psi_2$ 
       $[\phi']_M \leftarrow \{s \mid s \in S \wedge$ 
        UNTIL-STRATEGY( $node_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi\})$ 
      case  $\phi' = \langle\langle A^b \rangle\rangle \Box \psi$ 
       $[\phi']_M \leftarrow \{s \mid s \in S \wedge$ 
        BOX-STRATEGY( $node_0(s, b), \langle\langle A^b \rangle\rangle \Box \phi\})$ 
  return  $[\phi]_M$ 

```

Given ϕ , we produce a set of subformulas of ϕ $Sub(\phi)$ in the usual way, however in addition if $\langle\langle A^b \rangle\rangle \gamma \in Sub(\phi)$, its infinite resource version $\langle\langle A \rangle\rangle \gamma$ is added to $Sub(\phi)$. $Sub(\phi)$ is ordered in increasing order of complexity, in addition infinite resource versions of modal formulas come before bounded versions. Note that if a state s is not annotated with $\langle\langle A \rangle\rangle \gamma$ then s cannot satisfy the bounded resource version $\langle\langle A^b \rangle\rangle \gamma$.

We then proceed by cases. For all formulas in $Sub(\phi)$ apart from $\langle\langle A^b \rangle\rangle \bigcirc \phi$, $\langle\langle A^b \rangle\rangle \phi_1 \mathcal{U} \phi_2$ and $\langle\langle A^b \rangle\rangle \Box \phi$ we essentially run the standard ATL model-checking algorithm [2].

Labelling states with $\langle\langle A^b \rangle\rangle \bigcirc \phi$ makes use of a function $Pre^b(A, \rho)$ which, given a coalition A , a set $\rho \subseteq S$ and a bound b , returns a set of states s in which A has a joint action σ_A with $cost(s, \sigma_A) \leq b$ such that $out(s, \sigma_A) \subseteq \rho$. Labelling states with $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ is more complex, and in the interests of readability we provide separate functions: UNTIL-STRATEGY for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ formulas is shown in Algorithm 2, and BOX-STRATEGY for $\langle\langle A^b \rangle\rangle \Box \phi$ formulas is shown in Algorithm 3.

Both algorithms proceed by depth-first and-or search of M . We record information about the state of the search in a search tree of nodes. A *node* is a structure which consists of a state of M , the resources available to the agents A in that state (if any), and a finite path of nodes leading to this node from the root node. Edges in the tree correspond to joint actions by all agents. Note that the resources available to the agents in a state on a path constrain the edges from the corresponding node to be those actions σ_A where $cost(s, \sigma_A)$

is less than or equal to the available resources. For each node n in the tree, we have a function $s(n)$ which returns its state, $p(n)$ which returns the nodes on the path and $e_i(n)$ which returns the resource availability on the i -th resource in $s(n)$ as a result of following $p(n)$. The function $node_0(s, b)$ returns the root node, i.e., a node n_0 such that $s(n_0) = s$, $p(n_0) = []$ and $e_i(n_0) = b_i$ for all resources i . The function $node(n, a, s')$ returns a node n' where $s(n') = s'$, $p(n') = [p(n) \cdot n]$ and for all resources i , $e_i(n') = e_i(n) - c_i(a)$.

Algorithm 2 Labelling $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$

```

function UNTIL-STRATEGY( $n, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ )
  if  $s(n) \not\models \langle\langle A \rangle\rangle \phi \mathcal{U} \psi$  then
    return false
  if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \geq e_j(n))$  then
    return false
  if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \leq e_j(n)) \wedge$ 
     $e_i(n') < e_i(n)$  then
     $e_i(n) \leftarrow \infty$ 
  if  $s(n) \models \psi$  then
    return true
  if  $e(n) = \infty$  then
    return true
   $Act \leftarrow \{a \in Act(A, s(n)) \mid c(a) \leq e(n)\}$ 
  for  $a \in Act$  do
     $O \leftarrow$  states reachable by  $a$ 
     $strat \leftarrow true$ 
    for  $s' \in O$  do
       $strat \leftarrow strat \wedge$ 
        UNTIL-STRATEGY( $node(n, a, s'), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ )
    if  $strat$  then
      return true
  return false

```

Algorithm 3 Labelling $\langle\langle A^b \rangle\rangle \Box \phi$

```

function BOX-STRATEGY( $n, \langle\langle A^b \rangle\rangle \Box \phi$ )
  if  $s(n) \not\models \langle\langle A \rangle\rangle \Box \phi$  then
    return false
  if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') > e_j(n))$  then
    return false
  if  $\exists n' \in p(n) : s(n') = s(n) \wedge (\forall j : e_j(n') \leq e_j(n))$  then
    return true
   $Act \leftarrow \{a \in Act(A, s(n)) \mid c(a) \leq e(n)\}$ 
  for  $a \in Act$  do
     $O \leftarrow$  states reachable by  $a$ 
     $strat \leftarrow true$ 
    for  $s' \in O$  do
       $strat \leftarrow strat \wedge$ 
        BOX-STRATEGY( $node(n, a, s'), \langle\langle A^b \rangle\rangle \Box \phi$ )
    if  $strat$  then
      return true
  return false

```

Lemma 1. Algorithm 1 terminates.

Proof. All the cases in Algorithm 1 apart from $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ can be computed in time polynomial in $|M|$ and $|\phi|$. The cases for $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ and $\langle\langle A^b \rangle\rangle \Box \phi$ involve calling the UNTIL-STRATEGY and BOX-STRATEGY procedures, respectively, for every

state in S . We want to show that there is no infinite sequence of calls to UNTIL-STRATEGY or BOX-STRATEGY. Assume to the contrary that n_1, n_2, \dots is an infinite sequence of nodes in an infinite sequence of recursive calls to UNTIL-STRATEGY or BOX-STRATEGY. Then, since the set of states is finite, there is an infinite subsequence n_{i_1}, n_{i_2}, \dots of n_1, n_2, \dots such that $s(n_{i_j}) = s(n_{i_k})$. We show that there is an infinite subsequence n'_1, n'_2, \dots of n_{i_1}, n_{i_2}, \dots such that for $k < j$ $e(n'_k) \leq e(n'_j)$. Note that since n'_k and n'_j have the same state, both UNTIL-STRATEGY or BOX-STRATEGY will return in n'_j : a contradiction. The proof is very similar to the proof of Lemma f in [8, p.70] and proceeds by induction on the number of resources r . For $r = 1$, since $e(n)$ is always positive, the claim is immediate. Assume the lemma holds for r and let us show it for $r+1$. Then there is an infinite subsequence m'_1, m'_2, \dots of n_{i_1}, n_{i_2}, \dots where for all resources $i \in \{1, \dots, r\}$ $e_i(m'_k) \leq e_i(m'_j)$ for $k < j$. Clearly if we take m'_1 for the first element in the sequence of nodes with increasing resource availability we are constructing, there is a node m_j in the sequence m'_1, m'_2, \dots where $e_{r+1}(m'_1) \leq e_{r+1}(m'_j)$. We take m'_j to be n'_2 and repeat. \square

Before we prove correctness of UNTIL-STRATEGY and BOX-STRATEGY, we need some auxiliary notions. Let n be a node where one of the procedures returns true. We will refer to $tree(n)$ as the tree representing the successful call to the procedure. In particular, if the procedure returns true before any recursive calls are made, then $tree(n) = n$. Otherwise the procedure returns true because there is an action $\alpha \in Act$ such that for all $s' \in out(s(n), \alpha)$ the procedure returns true in $n' = node(n, \alpha, s')$. In this case, $tree(n)$ has n as its root and trees $tree(n')$ are the children of n . We refer to the action α as n_{act} (the action that generates the children of n). For the sake of uniformity, if $tree(n) = n$ then we set n_{act} to be *idle*. Such a tree corresponds to a strategy F where for each path $n \dots m$ from the root n to a node m in $tree(n)$, $F(s(n) \dots s(m)) = m_{act}$.

A strategy F for satisfying $\langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$ is \mathcal{U} -economical for a node n if, intuitively, no path generated by it contains a loop that does not increase any resource. A strategy is \square -economical for a node n if, intuitively, no path generated by it contains a loop that decreases some resources and does not increase any other resources. Formally, a strategy F is \mathcal{U} -economical for n if

- F satisfies $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ at $s(n)$, i.e., $\forall \lambda \in out(s(n), F)$, $\exists i \geq 0 : \lambda[i] \models \psi$ and $\lambda[j] \models \phi$ for all $j \in \{0, \dots, i\}$
- The path $p(n) \cdot n$ is already economical; i.e., $\forall n' \in p(n) \cdot n, n'' \in p(n') : s(n'') = s(n') \Rightarrow e(n'') \not\geq e(n')$;
- Every state is reached by F economically; i.e., $\forall s_0 s_1 \dots s_k \dots \in out(s(n), F)$ where $k \leq i$ and i is the first index in $s_0 s_1 \dots s_k \dots$ to satisfy ψ , $\forall j < k : s_j = s_k \Rightarrow cost(s_j \dots s_k) \not\geq 0$ where $cost(s_j \dots s_k) = \sum_{l=j, \dots, k-1} cost(\lambda[l], F(\lambda[0, l]))$; and
- Every state is reached by F economically with respect to the path $p(n)$; i.e., $\forall s_0 s_1 \dots s_k \dots \in out(s(n), F)$, $\forall n' \in p(n) : s(n') = s_k \Rightarrow e(n') \not\geq e(n) - cost(s_0 \dots s_k)$

A strategy F is \square -economical if:

- F satisfies $\langle\langle A^{e(n)} \rangle\rangle \square \phi$ at $s(n)$, i.e., $\forall \lambda \in out(s(n), F)$, $\forall i \geq 0 : \lambda[i] \models \phi$;
- The path $p(n) \cdot n$ is already economical; i.e., $\forall n' \in p(n) \cdot n, n'' \in p(n') : s(n'') = s(n') \Rightarrow e(n'') \not\geq e(n')$;
- Every state is reached by F economically; i.e., $\forall s_0 s_1 \dots s_k \dots \in out(s(n), F) \forall j < k : s_j = s_k \Rightarrow cost(s_j \dots s_k) \not\geq 0$;
- Every state is reached by F economically with respect to the path $p(n)$; i.e., $\forall s_0 s_1 \dots s_k \dots \in out(s(n), F)$, $\forall n' \in p(n) : s(n') = s_k \Rightarrow e(n') \not\geq e(n) - cost(s_0 \dots s_k)$.

Note that any strategy F satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ ($\langle\langle A^{e(n)} \rangle\rangle \square \phi$) at $s(n)$ can be converted to an economical one by eliminating unproductive loops.

Next we prove correctness of UNTIL-STRATEGY. The next lemma essentially shows that replacing a resource value with ∞ in Algorithm 2 is harmless. For the inductive proof, we need the following notion. Given a tree $tree(n)$ we call the result of removing all children of some nodes m_1, \dots, m_k that have only leaves as children in $tree(n)$, $(tree(n), prune(m_1, \dots, m_k))$ (or a pruning of $tree(n)$).

Lemma 2. Let $n = node_0(s, b)$ be a node where UNTIL-STRATEGY returns true. Let f be a function that for each leaf n' of $tree(n)$ returns $f(n') \in \mathbb{N}^r$ such that $f_i(n') = e_i(n')$ if $e_i(n') \neq \infty$. Then there is a strategy F which eventually generates at least $f(n')$ for all leaves n' of $tree(n)$.

Proof. (sketch) By induction on the structure of $tree(n)$.

Base Case: Let $tree(n)$ contain only its root. The proof is obvious for any strategy.

Inductive Step: Let us consider a pruning T' of $tree(n)$. By the induction hypothesis, any tree T' that has a less complex structure than T has a strategy to generate at least $f(n') \in \mathbb{N}^r \leq e(n')$ for all leaves n' of T' .

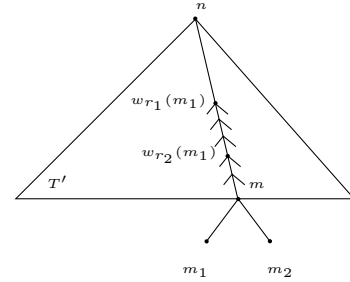


Figure 2. Tree T and $T' = (T, prune(m))$.

Let $m(m_1, \dots, m_k)$ be an arbitrary depth-1 sub-tree of T (see Figure 2). By removing $m(m_1, \dots, m_k)$ from T , we obtain a pruning T' of T .

Let $n \dots m \cdot m_i$ be a path in T from the root n to one of the leaves m_i . For each resource r the availability of which turns to ∞ at m_i , there must be a node $w_r(m_i)$ in the path $n \dots m \cdot m_i$ which is used to turn the availability of r to ∞ at m_i . We may repeat the path from $w_r(m_i)$ to m_i several times to generate enough resource availability for r . We call the path from $w_r(m_i)$ to m_i together with all the immediate child nodes of those along the path the column graph from $w_r(m_i)$ to m_i . Each time, an amount of $g_r = e_r(m) - c_r(a(m)) - e_r(w_r(m_i))$ is generated. Then, the minimal number of times to repeat the path from $w_r(m_i)$ to m_i is $h_r(m_i) = \lceil \frac{f_r(m_i) - e_r(m) - c_r(a(m))}{g_r} \rceil$.

Note that we need to repeat at each m_i for each resource r the path from $w_r(m_i)$ to m_i h_r times. To record the number of times the path has been repeated, we attach to each m_i a counter \hat{h}_r for each r and write the new node of m_i as $m_i^{\hat{h}_r(m_i)}$.

Initially, $\hat{h}_r = 0$ for all r . A step (see Figure 3) of the repetition is done as follows: let $m_i^{\hat{h}_r(m_i)}$ be some node such that $\hat{h}_r(m_i) < h_r(m_i)$. Let $m_j^{\hat{h}_r(m_j)}$ be the sibling of $m_i^{\hat{h}_r(m_i)}$ ($j \neq i$). We extend

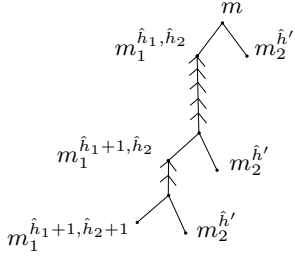


Figure 3. Repeating steps to generate resources.

from $m_i^{\hat{h}}$ the column-tree from $w_r(m_i)$ to m_i ; each new m_j ($j \neq i$) is annotated with $\hat{h}(m_j)$ (same as before) and the new m_i is annotated with $\hat{h}(m_i)$ except that $\hat{h}_r(m_i)$ is increased by 1. We repeat the above step until no further step can be made (it must terminate due to the fact that $\hat{h}_r(m_i) < \infty$ for all r and m_i).

At the end, we obtain a tree where all leaves $m_i^{\hat{h}}$ have $\hat{h}_r = h_r(m_i)$ for all r , hence the availability of r is at least f_r . Let $E(m)$ be the extended tree from m .

Let $F_{T'}$ be the generated strategy from T' . We extend $F_{T'}$ with $E(m)$ for every occurrence of m in $F_{T'}$ and denote this extended strategy $F_{T'}^E$. For all leaves m' in $E(m)$ which are other than m_i , let $E(m')$ be some sub-tree of T' which starts from m' . Then, we extend $F_{T'}^E$ with $E(m')$ for every occurrence of m' in $E(m)$. We finally obtain a tree F_T which satisfies the condition that all leaves l have resource availability of at least $f(l)$.

□

Corollary 1. If $\text{UNTIL-STRATEGY}(\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns true then $s \models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$.

Lemma 3. If $\text{UNTIL-STRATEGY}(n, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns false, then there is no \mathcal{U} -economical strategy from $s(n)$ satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$.

Proof. (sketch) We prove the lemma by induction on the depth of calling $\text{UNTIL-STRATEGY}(n, \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$.

Base Case: If false is returned by the first if-statement, then $s(n) \not\models \langle\langle A \rangle\rangle \phi \mathcal{U} \psi$; this also means there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$.

If false is returned by the second if-statement, then any strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$ is not economical.

Inductive Step: If false is not returned by the first two if-statements, then, for all actions $a \in \text{Act}$, there exists $s' \in \text{out}(s(n), a)$ such that $\text{UNTIL-STRATEGY}(n', \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ (where $n' = \text{node}(n, a, s')$) returns false. By induction hypothesis, there is no economical strategy satisfying $\langle\langle A^{e(n')} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n')$. Assume to the contrary that there is an economical strategy satisfying $\langle\langle A^{e(n')} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$. Let $a = F(s(n))$, then $a \in \text{Act}$. Obviously, for all $s' \in \text{out}(s(n), a)$, $F'(\lambda) = F(s(n)\lambda)$ is an economical strategy from $n' = \text{node}(n, a, s')$. This is a contradiction; hence, there is no economical strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \phi \mathcal{U} \psi$ from $s(n)$.

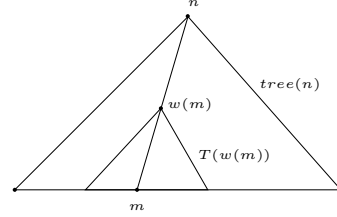
□

Corollary 2. If $\text{UNTIL-STRATEGY}(\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi)$ returns false then $s \not\models \langle\langle A^b \rangle\rangle \phi \mathcal{U} \psi$.

Now we turn to Algorithm 3 for labelling states with $\langle\langle A^b \rangle\rangle \Box \phi$. First we show the soundness of Algorithm 3.

Lemma 4. Let $n = \text{node}_0(s, b)$. If $\text{BOX-STRATEGY}(n, \langle\langle A^b \rangle\rangle \Box \phi)$ returns true then $s(n) \models \langle\langle A^b \rangle\rangle \Box \phi$.

Proof. (sketch) In the following, for each node m in $\text{tree}(n)$, let $T(m)$ denote the sub-tree of $\text{tree}(m)$ rooted at m . For each leaf

Figure 4. $w(m)$ of m in $\text{tree}(n)$.

m of $\text{tree}(n)$, let $w(m)$ denote one of the nodes in $p(m)$ such that $s(w(m)) = s(m)$ and $e(w(m)) \leq e(m)$ (see Figure 4).

Let us expand $\text{tree}(n)$ as follows:

- T^0 is $\text{tree}(n)$;
- T^{i+1} is T^i where all its leaves m are replaced by $T(w(m))$ (see Figure 5);

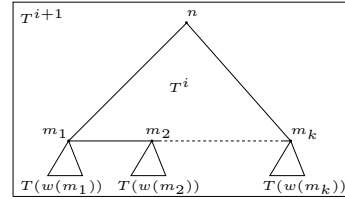


Figure 5. One step in constructing the strategy.

Let $T = T^\infty$, then T is a strategy for $\langle\langle A^b \rangle\rangle \Box \phi$.

□

Lemma 5. If $\text{BOX-STRATEGY}(n, \langle\langle A^b \rangle\rangle \Box \phi)$ returns false, then there is no \Box -economical strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ at $s(n)$.

Proof. (sketch) The proof is done by induction on the depth of calling $\text{BOX-STRATEGY}(n, \langle\langle A^b \rangle\rangle \Box \phi)$.

Base Case: If false is returned by the first if-statement, then $s(n) \not\models \langle\langle A \rangle\rangle \Box \phi$; this also means there is no strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ at $s(n)$.

If false is returned by the second if-statement, then any strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ at $s(n)$ is not \Box -economical.

Inductive Step: If false is not returned by the first two if-statements, for all actions $a \in \text{Act}$, there exists $s' \in \text{out}(s(n), a)$ such that $\text{BOX-STRATEGY}(n', \langle\langle A^b \rangle\rangle \Box \phi)$ (where $n' = \text{node}(n, a, s')$) returns false. Assume to the contrary that there is a \Box -economical strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ from $s(n)$. Let $a = F(s(n))$, then $a \in \text{Act}$. Obviously, for all $s' \in \text{out}(s(n), a)$, $F'(\lambda) = F(s(n)\lambda)$ is a \Box -economical strategy from $n' = \text{node}(n, a, s')$. This is a contradiction; hence, there is no \Box -economical strategy satisfying $\langle\langle A^{e(n)} \rangle\rangle \Box \phi$ from $s(n)$.

□

Corollary 3. If $\text{BOX-STRATEGY}(\text{node}_0(s, b), \langle\langle A^b \rangle\rangle \Box \phi)$ returns false then $s \not\models \langle\langle A^b \rangle\rangle \Box \phi$.

4 LOWER BOUND

In this section we show that the lower bound for the complexity of the model checking problem for $\text{RB}\pm\text{ATL}$ is EXPSPACE , by reducing from the reachability problem of Petri Nets. Note that the exact complexity of this problem is still an open question (although it is known to be decidable, [8]), hence the same holds for the exact complexity of the $\text{RB}\pm\text{ATL}$ model-checking problem.

A Petri net is a tuple $N = (P, T, W, M)$ where:

- P is a finite set of places;
- T is a finite set of transitions;
- $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ is a weighting function; and
- $M : P \rightarrow \mathbb{N}$ is an initial marking.

A transition $t \in T$ is enabled iff $W(r, t) \leq M(r)$ for all $r \in P$. The result of performing t is a marking M' where $M'(r) = M(r) - W(r, t) + W(t, r)$, denoted as $M[t] M'$.

A marking M' is reachable from M iff there exists a sequence

$$M_0 [t_1] M_1 [t_2] \dots [t_n] M_n$$

where $M_0 = M$ and $n \geq 0$ such that $M_n \geq M$ (where $M \geq M'$ iff $M(r) \geq M'(r)$ for all $r \in P$). It is known that the lower bound for the complexity of this version of the reachability problem (with $M_n \geq M$ rather than $M_n = M$) is EXPSPACE [8, p.73].

We present a reduction from an instance of the reachability problem of Petri Nets to an instance of the model checking problem of $\text{RB}\pm\text{ATL}$.

Given a net $N = (P, T, W, M)$ and a marking M' , we construct a $\text{RB-CGS } I_{N, M'} = (\{1\}, P, S, \{p\}, \pi, \text{Act}, d, c, \delta)$ where:

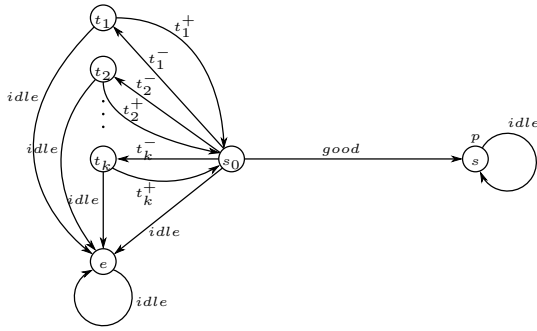


Figure 6. Structure $I_{N, M'}$.

- $S = \{s_0\} \cup T \cup \{s, e\}$;
- $\pi(p) = \{s\}$;
- $\text{Act} = \{\text{idle}, \text{good}\} \cup \{t^-, t^+ \mid t \in T\}$;
- $d(s_0) = \{\text{idle}, \text{good}\} \cup \{t^- \mid t \in T\}$;
- $d(s) = d(e) = \{\text{idle}\}$;
- $d(t) = \{\text{idle}, t^+\}$;
- $c(\text{idle}) = \bar{0}$; $c(\text{good}) = M'$;
- $c_r(t^-) = W(r, t)$ for all $r \in P$;
- $c_r(t^+) = -W(r, t)$ for all $r \in P$;
- $\delta(x, \text{idle}) = e$ for $x \in \{s_0, t, e\}$;

- $\delta(s_0, \text{good}) = s$;
- $\delta(s_0, t^-) = t$;
- $\delta(t, t^+) = s_0$.

The following is straightforward:

Lemma 6. Given a net $N = (P, T, W, M)$ and a marking M' , M' is reachable from M iff $I_{N, M'}, s_0 \models \langle\langle 1^M \rangle\rangle \top \mathcal{U} p$.

Corollary 4. The lower bound for the model checking problem complexity of $\text{RB}\pm\text{ATL}$ is EXPSPACE .

5 CONCLUSION

The main contribution of this paper is a model-checking algorithm for $\text{RB}\pm\text{ATL}$, a logic with *resource production*. This is the first decidability result for a resource logic of strategic ability (multi-agent rather than single agent) that allows both unbounded production and consumption of resources. The lower bound for the model-checking complexity of $\text{RB}\pm\text{ATL}$ is EXPSPACE and the upper bound is still an open problem. In future work, we plan to concentrate on identifying computationally tractable cases for $\text{RB}\pm\text{ATL}$ model-checking, for example by restricting the class of transition systems to those without ‘mixed’ loops (producing one resource and consuming another).

Acknowledgments This work was supported by the Engineering and Physical Sciences Research Council [grants EP/K033905/1 and EP/K033921/1]. We would also like to thank the anonymous ECAI 2014 reviewers whose comments and suggestions helped to improve the paper.

REFERENCES

- [1] N. Alechina, B. Logan, H. N. Nguyen, and A. Rakib, ‘Resource-bounded alternating-time temporal logic’, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pp. 481–488. IFAAMAS, (2010).
- [2] R. Alur, T. Henzinger, and O. Kupferman, ‘Alternating-time temporal logic’, *Journal of the ACM*, **49**(5), 672–713, (2002).
- [3] N. Bulling, J. Dix, and W. Jamroga, ‘Model checking logics of strategic ability: Complexity*’, in *Specification and Verification of Multi-agent Systems*, 125–159, Springer, (2010).
- [4] N. Bulling and B. Farwer, ‘On the (un-)decidability of model checking resource-bounded agents’, in *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010)*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 567–572. IOS Press, (2010).
- [5] E. M. Clarke, E. A. Emerson, and A. P. Sistla, ‘Automatic verification of finite-state concurrent systems using temporal logic specifications’, *ACM Transactions on Programming Languages and Systems*, **8**(2), 244–263, (1986).
- [6] D. Della Monica, M. Napoli, and M. Parente, ‘On a logic for coalitional games with priced-resource agents’, *Electr. Notes Theor. Comput. Sci.*, **278**, 215–228, (2011).
- [7] D. Della Monica, M. Napoli, and M. Parente, ‘Model checking coalitional games in shortage resource scenarios’, in *Proceedings of the 4th International Symposium on Games, Automata, Logics and Formal Verification (GandALF 2013)*, volume 119 of *EPTCS*, pp. 240–255, (2013).
- [8] W. Reisig, *Petri Nets: An Introduction*, volume 4 of *EATCS Monographs on Theoretical Computer Science*, Springer, 1985.